Theoretical Computer Science 449 (2012) 54–63

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Geometrical regular languages and linear Diophantine equations: The strongly connected case

Jean-Marc Champarnaud [a,*], Jean-Philippe Dubernard [a], Franck Guingne [b], Hadrien Jeanne [a]

[a] LITIS, Université de Rouen, France

[b] I3S, Université de Nice - Sophia Antipolis & CNRS, France

A R T I C L E I N F O

Keywords:
Regular language
Minimal deterministic automaton
Parikh image
Geometrical language
Diophantine equation

A B S T R A C T

Given an arbitrarily large alphabet Σ, we consider the family of regular languages over Σ for which the deterministic minimal automaton has a strongly connected state diagram. We present a new method for checking whether such a language is semi-geometrical or not and whether it is geometrical or not. This method makes use of the enumeration of the simple cycles of the state diagram. It is based on the construction of systems of linear Diophantine equations, where the coefficients are deduced from the set of simple cycles.

1. Introduction

A d-dimensional geometrical figure is a (possibly infinite) set of sites of the d-dimensional oriented site square lattice connected by a network of nearest neighbour bonds. It turns out that a finite geometrical figure is an animal [12]. The main interest of the study of the d-dimensional geometrical figures comes from the problem of tiling such a figure by an animal. This problem is close to the problem of tiling the cell lattice by a polyomino [12], which has been much studied. An equivalent definition of a geometrical figure is given in [1], where an application to the modelling of real-time task systems [11] is mentioned.

Let $\Sigma = \{a_1, \ldots, a_d\}$ be an ordered alphabet with d symbols. The Parikh mapping [21] $c : \Sigma^* \mapsto \mathbb{N}^d$ maps a word $w \in \Sigma^*$ to its coordinate vector $(|w|_{a_1}, \ldots, |w|_{a_d}) \in \mathbb{N}^d$, which allows us to transform a geometrical figure F into a labelled graph: if P and Q are neighbours w.r.t. the kth direction, with $1 \leq k \leq d$, *i.e.* if $Q = P + c(a_k)$, then there exists a labelled arc (P, a_k, Q) in the graph. As a consequence, the geometrical figure of a prefix closed language is the set of the Parikh images of its words. Conversely, the language of a geometrical figure is the set of the words that are labels of a path going from the origin in the figure. Any prefix closed language is a subset of the language of its geometrical figure, but the reciprocal is false. This property leads to the definition of two sub-families of prefix closed languages: a geometrical language is equal to the language of its geometrical figure; a semi-geometrical language is such that two words with the same Parikh image define identical left residuals.

The study of geometrical figures is facilitated by making use of the properties of their languages. For example, the problem of tiling a geometrical figure by an animal is easier to solve if the figure is generated by a regular language [5]. However, even for the regular case, there exists no polynomial-time algorithm for checking whether a language is semi-geometrical or not and whether it is geometrical or not, except for $d \leq 2$ (see [4]).

In this paper, we consider the family of regular languages for which the deterministic minimal automaton has a strongly connected state diagram. Our aim is to design a new method for checking whether such a language, over an arbitrarily large alphabet, is semi-geometrical or not and whether it is geometrical or not.

* Corresponding author. Tel.: +33 02 32 95 53 05.
E-mail addresses: Jean-Marc.Champarnaud@univ-rouen.fr (J.-M. Champarnaud), jean-philippe.dubernard@univ-rouen.fr (J.-P. Dubernard), franck.guingne@i3s.unice.fr (F. Guingne), hadrien.jeanne@univ-rouen.fr (H. Jeanne).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.04.012

This method is firstly based on the enumeration of the simple cycles of a strongly connected directed graph [24] and secondly on the construction of a set of systems of linear Diophantine equations [16], where the coefficients are deduced from the set of simple cycles. For both tests (is the language semi-geometrical? is the language geometrical?), the answer depends on the existence or not of a solution for such systems.

As far as complexity is concerned, the new method involves two non polynomial-time steps: the NFA determinization, at least if the data is a regular expression denoting the language, and the enumeration of the simple cycles of a strongly connected graph. Moreover, it is based on the resolution of a set of systems of linear Diophantine equations. However, the only method independent of the size of the alphabet known until now [1] presents a similar drawback since it is based on two non polynomial-time steps: the conversion of a finite automaton into a regular expression (whose worst case size is exponential) and the handling of Parikh coordinate sets [1]. Beyond the theoretical interest of this new method, the hope is that powerful tools to solve linear Diophantine equation systems, such as the Polylib library [25] or the LinBox library [13], would yield good results. Notice that an approach of a similar nature has been successfully developed in [10], where a SAT solver is used to attack the NFA reduction problem.

The next section recalls the notation and definitions concerning languages, automata and geometrical figures. Section 3 contains a study of the set of paths of a d-ary strongly connected graph and Section 4 is devoted to the design of a semi-geometricity test and of a geometricity test inside such a graph. The last section reports some experimental results.

2. Preliminaries

2.1. Languages, automata and graphs

Let us first review basic notions concerning regular languages and finite automata. For a comprehensive treatment of this domain, reference [9] can be consulted. Let Σ be a nonempty finite set of symbols, called the *alphabet*. An alphabet is said to be *ordered* if it is equipped with an order relation. A *word* over Σ is a finite sequence of symbols, usually written $x_1x_2\cdots x_n$. The *length* of a word u, denoted by $|u|$, is the number of symbols in u. The number of occurrences of a symbol a in u is denoted by $|u|_a$. The *empty word*, denoted by ε, has a null length. If $u = x_1\cdots x_n$ and $v = y_1\cdots y_m$ are two words over the alphabet Σ, their concatenation $u \cdot v$, usually written uv, is the word $x_1\cdots x_n y_1\cdots y_m$. Let Σ^* be the set of words over Σ. Given two words u and w in Σ^*, u is said to be a *prefix* of w if there exists a word v in Σ^* such that $uv = w$. A *language* L over Σ is a subset of Σ^*. The *left residual* of a language L w.r.t. a word $u \in \Sigma^*$ is the set $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$. The prefix closure of a language L is the set Pref(L) of the prefixes of its words. A language is said to be *prefix closed* if it is equal to its prefix closure. *Regular languages* over an alphabet Σ are the smallest family of subsets of Σ^* that contains the empty set and the set $\{a\}$ for all $a \in \Sigma$ and that is closed under concatenation, union and star.

A *deterministic finite automaton* (DFA) is a 5-tuple $\mathcal{A} = (Q, \Sigma, \delta, s_0, T)$ where Q is a finite nonempty *set of states*, δ is a mapping from $Q \times \Sigma$ to Q, $s_0 \in Q$ is the *initial state* and $T \subseteq Q$ is the *set of final states*. For all $(p, x) \in Q \times \Sigma$, we will write $p \cdot x$ instead of $\delta(p, x)$; for all $Q' \subset Q$ and for all $x \in \Sigma$, the set $\{p \cdot x \mid p \in Q'\}$ is denoted by $Q' \cdot x$. The 3-tuple (p, x, q) in $Q \times \Sigma \times Q$ is said to be *a transition* if and only if $q = p \cdot x$. A DFA $\mathcal{A}$ is said to be *complete* if for any $q \in Q$ and any $a \in \Sigma$, $|q \cdot a| = 1$. In a complete DFA there may exist a *sink state* σ such that $\sigma \notin T$ and, for all $x \in \Sigma$, $\sigma \cdot x = \sigma$.

The *state diagram* of the automaton $\mathcal{A} = (Q, \Sigma, \delta, s_0, T)$ is the *graph* $\mathcal{G} = (Q, U, \Sigma)$, where Q is the set of *vertices* and $U \subset Q \times \Sigma \times Q$ is the set of *labelled arcs*, with $U = \delta$. We say that $(p, q) \in Q \times Q$ is an *arc* if there exists $a \in \Sigma$ such that $(p, a, q) \in U$. By definition the state diagram of an automaton is a *directed* graph and a *labelled* one. Let $d = |\Sigma|$. The automaton $\mathcal{A}$ and the state diagram $\mathcal{G}$ are said to be *unary* if $d = 1$, *binary* if $d = 2$ and *d-ary* otherwise.

The following notions are actually defined on the state diagram $\mathcal{G}$ of the DFA. Let p and q be two states in Q. A *path* going from p to q and labelled by the word $u = u_1\cdots u_t \in \Sigma^*$ is the sequence of states $(p_0 = p, p_1 = p_0{\cdot}u_1, \ldots, p_t = p_{t-1}{\cdot}u_t = q)$. The state p (resp. q) is said to be *the head* (resp. *the tail*) of the path. The word u is said to be the *label* of the path and its length t is the length of the path. A *cycle* is a path where the head and the tail are the same state. A path (resp. a cycle) $(p_0 = p, p_1 = p_0 \cdot u_1, \ldots, p_t = p_{t-1} \cdot u_t)$ is said to be *simple* if the states $p_0, \ldots, p_t$ (resp. $p_0, \ldots, p_{t-1}$) are pairwise distinct. A DFA $\mathcal{A}$ is said to be *accessible* if for any $q \in Q$ there exists a path from s_0 to q. A DFA is said to be *strongly connected* if for all $(i, j) \in Q \times Q$ with $i \neq j$, there exist a path going from i to j and a path going from j to i.

A path starting from a state p and labelled by a word $u \in \Sigma^*$ is said to be *successful* if $p = s_0$ and $p \cdot u \in T$. The language $L(\mathcal{A})$ *recognized* by the DFA $\mathcal{A}$ is the set of words that are labels of successful paths. Kleene's theorem [15] states that a language is recognized by a DFA if and only if it is regular. The *left language* $\overleftarrow{L}_q^{\mathcal{A}}$ (resp. *right language* $\overrightarrow{L}_q^{\mathcal{A}}$) of a state q is the set of words w such that there exists a path in $\mathcal{A}$ from s_0 to the state q (resp. from q to a final state) with w as label. A complete and accessible DFA $\mathcal{A}$ is *minimal* if and only if any two distinct states of $\mathcal{A}$ have distinct right languages. According to the theorem of Myhill–Nerode [18,19], the minimal DFA of a regular language is unique up to an isomorphism.

2.2. Geometrical figures and their languages

We consider the d-dimensional oriented site square lattice (site lattice for short).

Definition 1. *A* d-dimensional geometrical figure F *is a (possibly infinite) set of sites of the site lattice connected by a network of nearest neighbour bonds.*

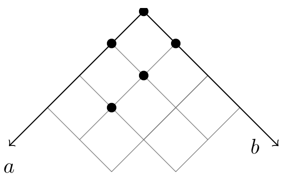

Fig. 1. $F_1 = \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 1)\}$.

We denote by $\mathcal{O}$ the point with coordinate $(0, \ldots, 0)$. The *level* of the point $P = (x_1, \ldots, x_d)$ is $\mathrm{level}(P) = x_1 + \cdots + x_d$. Notice that 2-dimensional geometrical figures (such as the one of Figure 1) are drawn so that two points with the same level lie on a same horizontal line.

Let $F \subset \mathbb{N}^d$ be a d-dimensional geometrical figure. With F can be associated the directed graph $G = (F, U)$ such that the set F of sites or *points* is the set of *vertices* of G and the set $U \subset F \times F$ is the set of the implicit *arcs* that are defined by the only d authorized bonds from a point of F to a neighbouring one (that are the South-East and South-West steps for the 2-dimensional figure of Fig. 1). It is easy to make F be a labelled figure by considering an ordered alphabet $\Sigma = \{a_1, \ldots, a_d\}$ of d symbols each associated with a direction of the site lattice. More precisely, the Parikh mapping [21] $c : \Sigma^* \mapsto \mathbb{N}^d$ maps a word $w \in \Sigma^*$ to its coordinate vector $c(w) = (|w|_{a_1}, \ldots, |w|_{a_d})$. Hence the following definition: the *geometrical figure of a prefix closed language L* is the figure $\mathcal{F}(L) = \bigcup_{w \in L} c(w)$.

In a geometrical figure, a *trajectory* (P, u), with $P \in F$ and $u = u_1 \cdots u_p \in \Sigma^*$ is a sequence $(P_0 = P, P_1, \ldots, P_p)$ of points in F such that for all $1 \le i \le p$, $P_i = P_{i-1} + c(u_i)$. Let $\mathrm{Traj}(\mathcal{O}, F)$ be the set of trajectories of F starting from $\mathcal{O}$. The *language of a geometrical figure F* is the language $\mathcal{L}(F) = \{u \mid (\mathcal{O}, u) \in \mathrm{Traj}(\mathcal{O}, F)\}$. For any prefix closed language L, it holds that $L \subseteq \mathcal{L}(\mathcal{F}(L))$. Some languages however are such that $\mathcal{L}(\mathcal{F}(L)) \nsubseteq L$. For instance, the two languages $L_1 = \{\varepsilon, a, b, ba\}$ and $L_2 = \{\varepsilon, a, b, ab, ba\}$ over the alphabet $\Sigma = \{a, b\}$ have the same geometrical figure F; it can be checked that $\mathcal{L}(F) = L_2$ whereas $\mathcal{L}(F) \nsubseteq L_1$ since ab is in $\mathcal{L}(F)$ but is not in L_1. Hence we give the definition of the two following families of languages:

Definition 2. *A prefix closed language L is* geometrical *if* $L = \mathcal{L}(\mathcal{F}(L))$.

Definition 3. *A prefix closed language L is* semi-geometrical *if, for all u and v in L, the condition* $c(u) = c(v) \Rightarrow u^{-1}L = v^{-1}L$ *is satisfied.*

Definition 4. *A regular language L is said to be* geometrical *(resp.* semi-geometrical*) if* $\mathrm{Pref}(L)$ *is geometrical (resp. semi-geometrical).*

2.3. Arithmetic

Given two integers a and b we write $a|b$ if a is a divisor of b. The greatest common divisor of n integers $a_1, \ldots, a_n$ is denoted by $\gcd(a_1, \ldots, a_n)$.

Theorem 1 (*Brauer Theorem [2]*)**.** *Let r be a positive integer and* $l_1, l_2, \ldots, l_r$ *be r integers such that* $0 < l_1 < l_2 < \cdots < l_r$ *and* $\gcd(l_1, l_2, \ldots, l_r) = 1$. *Let us set* $m_0 = (l_1 - 1)(l_r - 1)$. *Then every integer* $m \ge m_0$ *is a linear combination with non-negative coefficients of the integers* $l_1, \ldots, l_r$.

If the integers $l_1, \ldots, l_r$ are not mutually prime, let us set $p = \gcd(l_1, l_2, \ldots, l_r)$. Brauer theorem can be applied to the r integers $\frac{l_1}{p}, \frac{l_2}{p}, \ldots, \frac{l_r}{p}$ that are such that $0 < \frac{l_1}{p} < \frac{l_2}{p} < \cdots < \frac{l_r}{p}$ and $\gcd(\frac{l_1}{p}, \frac{l_2}{p}, \ldots, \frac{l_r}{p}) = 1$.

Corollary 1. *Let r be a positive integer and* $l_1, l_2, \ldots, l_r$ *be r integers such that* $0 < l_1 < l_2 < \cdots < l_r$ *and* $p = \gcd(l_1, l_2, \ldots, l_r)$. *Let us set* $m_0 = (\frac{l_1}{p} - 1)(\frac{l_r}{p} - 1)$. *Then for all integer* $m \ge m_0$, *mp is a linear combination with non-negative coefficients of the integers* $l_1, \ldots, l_r$.

The two following corollaries address the case where the r integers are not necessarily distinct or not necessarily non null. Since they are straightforward consequences of Corollary 1, the proof is omitted.

Let us consider now r integers $l_1, l_2, \ldots, l_r$ that are not necessarily distinct.

Corollary 2. *Let r be a positive integer and* $l_1, l_2, \ldots, l_r$ *be r integers. We assume that there exists an integer i such that* $1 \le i \le r - 1$ *and* $0 < l_1 < l_2 < \cdots < l_i = l_{i+1} < \cdots < l_r$. *Let* $p = \gcd(l_1, l_2, \ldots, l_r)$ *and* $m_0 = (\frac{l_1}{p} - 1)(\frac{l_r}{p} - 1)$. *Then for all integer* $m \ge m_0$, *mp is a linear combination with non-negative coefficients of the integers* $l_1, \ldots, l_r$.

Corollary 3. *Let r be a positive integer,* $l_1, l_2, \ldots, l_r$ *be r non-negative integers, and* $(l_{i_1}, \ldots, l_{i_s})$, *with* $s \le r$, *a sub-sequence of positive integers of the sequence* $(l_1, \ldots, l_r)$. *Let* $p = \gcd(l_{i_1}, \ldots, l_{i_s})$ *and* $m_0 = (\frac{l_{i_1}}{p} - 1)(\frac{l_{i_s}}{p} - 1)$. *Then for all integer* $m \ge m_0$, *mp is a linear combination with non-negative coefficients of the integers* $l_1, \ldots, l_r$.

Definition 5. *A* system of linear Diophantine equations *is a system* $Ax = b$, *where* $A = (a_{i,j})$ *is an* $m \times n$ *matrix with integer entries, b is an* $m \times 1$ *column vector with integer components and x is an* $n \times 1$ *solution vector with integer components.*

As far as the resolution of a linear Diophantine equation $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ is concerned, a well-known result is that solutions exist if and only if $\gcd(a_1, a_2, \ldots, a_n)|b$. This result admits extensions to the case of Diophantine linear systems [16], such as the following characterization: a Diophantine system $Ax = b$, where A is a matrix of m rows, has a solution x if and only if each subdeterminant of order m of the matrix $[A, b]$ is an integral multiple of the gcd of the subdeterminants of the matrix A of order m. The problem of finding any or all of the solutions of a system of linear Diophantine equations is solved thanks to linear algebra tools such as the Hermite normal form or the Smith normal form of a matrix [23]. However, there exists no polynomial-time algorithm straightforwardly deduced from the computation of these normal forms.

3. Strongly connected graphs

Our aim is to characterize the geometrical properties of a prefix closed regular language L through the properties of its minimal deterministic automaton $\mathcal{A}$, under the assumption that the state diagram $\mathcal{G} = (Q, U, \Sigma)$ of $\mathcal{A}$ is strongly connected. It turns out that these properties can be formulated in terms of graphs. We therefore first study some graph properties. More precisely, we assume that $\mathcal{G}$ is a d-ary strongly connected graph and, given two distinct vertices i and j, we focus on the set of paths going from i to j.

This study is a generalization of results that are described in [14,6] and used in [7] in the frame of unary strongly connected graphs. The case of strongly connected graphs is important for two reasons: first it is the basic case in the study of general graphs, and secondly, for a large enough d, the ratio of accessible n-state DFAs over an alphabet with d symbols having a strongly connected state diagram is conjectured to be asymptotically equal to 1.

Let $D_{(d,n)}$ be the number of accessible and complete n-state DFAs over an alphabet with d symbols and $F_{(d,n)}$ be the number of such DFAs having a strongly connected state diagram. Then, the following conjecture has been stated:

Proposition 4. *[22] The two following conditions (are conjectured to) asymptotically hold:*

$$D_{(d,n)} = n^{dn}\gamma_d^n(1 + o(1)),$$

with $\gamma_d = \frac{(1-c_d)^{\frac{1-c_d}{c_d}}}{c_d^{d-1}}$ *and* c_d *is the root of the equation* $c_d = 1 - e^{-dc_d}$, *and*

$$F_{(d,n)} \equiv c_d D_{(d,n)}.$$

As reported in [20], it holds that $c_2 = 0.7968$, $c_3 = 0.9405$ and $1 - 10^{-10} < c_{24} < 1$. Hence, for $d > 24$, the ratio of accessible n-state DFAs over an alphabet with d symbols having a strongly connected state diagram is asymptotically equal to 1.

In order to shorten notation, we consider an alphabet $\Sigma = \{1, \ldots, d\}$. Given a symbol $x \in \Sigma$, the vertex j is said to be a *x-successor* of the vertex i if there exists a labelled arc (i, x, j) in U. By abuse of notation the set of x-successors of a set of vertices $X \subset Q$ is denoted by $X \cdot x$.

Let $\mathcal{C}$ be the set of simple cycles of $\mathcal{G}$ and r the number of elements of $\mathcal{C}$. Any sequence $((\alpha_i, c_i))_{1\leq i\leq r}$ with, for all $1 \leq i \leq r$, $\alpha_i \in \mathbb{N}$ and $c_i \in \mathcal{C}$ is said to be a *combination of simple cycles*. Any cycle is a combination of simple cycles. Let i and j be two vertices of Q. The set of paths going from i to j is denoted by $\Gamma_{i,j}$. A path γ in $\Gamma_{i,j}$ can also be denoted by $\gamma(i, j)$ in order to point out its head and its tail.

Let γ be a path of $\Gamma_{i,j}$ and u be the label of γ. Besides the length of γ that is equal to the length $|u|$ of the word u, we define the following notions.

Definition 6. (1) *For all $x \in \Sigma$, the* x-length l^x *of the path γ labelled by $u \in \Sigma^*$ is the number $|u|_x$ of occurrences of x in the word u.*
(2) *The* Parikh vector $l(\gamma)$ *of the path γ is the vector $(l^1, \ldots, l^d)$ of $\mathbb{N}^d$. It will be denoted by l if there is no ambiguity.*

For all $c_k \in \mathcal{C}$, and for all $x \in \Sigma$, the x-length of the cycle c_k is denoted by l_k^x. The Parikh vector of the cycle c_k is the vector $l_k = (l_k^1, \ldots, l_k^d)$ of $\mathbb{N}^d$. Notice that for a unary graph, the Parikh vector of a path has only one component that is equal to the length of the path.

Let $a = (a_1, \ldots, a_d)$, $b = (b_1, \ldots, b_d)$ and $r = (r_1, \ldots, r_d)$ be three elements of $\mathbb{N}^d$. The following relations are defined on $\mathbb{N}^d$:

$$\begin{aligned}
&a \geq b \Leftrightarrow \forall x \mid 1 \leq x \leq d,\ a_x \geq b_x\\
&a \not\geq b \Leftrightarrow \exists x \mid 1 \leq x \leq d,\ a_x < b_x\\
&r = \max(a, b) \Leftrightarrow \forall x \mid 1 \leq x \leq d,\ r_x = \max(a_x, b_x)\\
&a = r \pmod{b} \Leftrightarrow \forall x \mid 1 \leq x \leq d,\ a_x = r_x \pmod{b_x}.
\end{aligned}$$

We say that a is *greater* (*not greater*) than b if $a \geq b$ (resp. $a \not\geq b$). We say that r is *the maximum* of a and b if $r = \max(a, b)$.

Definition 7. *Given two distinct vertices i and j, any path $\gamma \in \Gamma_{i,j}$ going through at least one vertex of every simple cycle of $\mathcal{G}$ is said to be a* basic path.

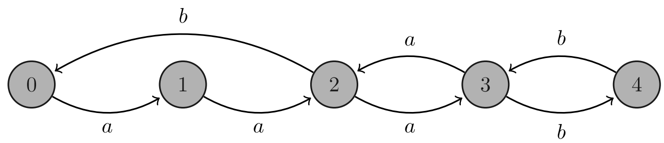

Fig. 2. The paths $\gamma_0(0, 2) = (0, 1, 2, 3, 2)$ and $\gamma_0(0, 3) = (0, 1, 2, 3)$ are basic paths.

The notion of a basic path is illustrated by Fig. 2. Let us remark that any path $\gamma \in \Gamma_{i,j}$ going through each vertex of Q at least once is a basic path. For instance $\gamma_0'(0, 2) = (0, 1, 2, 3, 4, 3, 2)$ and $\gamma_0'(0, 3) = (0, 1, 2, 3, 4, 3)$ are basic paths going through each vertex of Q at least once.

Lemma 5. *Let $\mathcal{G} = (Q, U, \Sigma)$ be a d-ary strongly connected graph and i and j two distinct vertices of Q.*
(1) *There exists a basic path $\gamma_0 \in \Gamma_{i,j}$.*
(2) *The path obtained by adding to the path γ_0 any combination of simple cycles of $\mathcal{G}$ belongs to $\Gamma_{i,j}$.*

Proof. (1) It is a straightforward consequence of the fact that $\mathcal{G}$ is strongly connected.
(2) Since the path γ_0 goes through at least one vertex of every simple cycle of $\mathcal{G}$, it is possible to augment it by any combination of simple cycles of G. The resulting path is obviously in $\Gamma_{i,j}$. $\square$

For any simple cycle in $\mathcal{G}$, its length is different from 0 and its Parikh vector is different from the null vector. In contrast, given a symbol $x \in \Sigma$, the x-length of a simple cycle can be null. However, since $\mathcal{G}$ is strongly connected, for every letter x in the input alphabet Σ of the automaton, there is at least one simple cycle with a non null x-length. Hence the following definition of period is useful.

Definition 8. *Let $x \in \Sigma$. Let us consider the sequence $(l_1^x, \ldots, l_r^x)$, where l_i^x, $1 \le i \le r$, is the x-length of the simple cycle c_i, and the sub-sequence $(t_k^x)_{1\le k\le s}$ of its non-null elements.*
Then we have: (1) *The x*-period p_x *of $\mathcal{G}$ is equal to* $\gcd(t_1^x, \ldots, t_s^x)$.
(2) *The* period p *of $\mathcal{G}$ is the vector $(p_1, \ldots, p_d)$ of $\mathbb{N}^d$.*

Lemma 6. *Let i and j be two vertices in Q. Then the Parikh vectors of the paths of $\Gamma_{i,j}$ are all equal modulo p.*

Proof. The Parikh vector of any cycle of $\mathcal{G}$ is a linear combination with non-negative coefficients of the Parikh vectors $l_1, \ldots, l_r$. Consequently, for all $x \in \Sigma$, the x-length of any cycle of $\mathcal{G}$ is a multiple of p_x. Let i and j be two distinct vertices of Q and let γ and γ' be two paths in $\Gamma_{i,j}$. Since $\mathcal{G}$ is strongly connected, there exists a path $\gamma'' \in \Gamma_{j,i}$. The paths $\gamma\gamma''$ and $\gamma'\gamma''$ are cycles that go through i. Let l (resp. l', l'') be the Parikh vector of the path γ (resp. γ', γ''). For all $x \in \Sigma$, the condition $l^x + l''^x = l'^x + l''^x \pmod{p_x}$ is satisfied and consequently $l^x = l'^x \pmod{p_x}$. $\square$

For all $x \in \Sigma$, the set of x-lengths of paths of $\Gamma_{i,j}$ is denoted by $\Lambda_x(i, j)$. We consider the subset $\Lambda_x^{+t}(i, j)$ (resp. $\Lambda_x^{-t}(i, j)$ of x-lengths not less than (resp. less than) a threshold t. The next proposition shows that there exists, for all $x \in \Sigma$, a threshold h_x beyond which the set $\Lambda_x(i, j)$ has a periodic behaviour. The threshold vector is denoted by $h = (h_1, \ldots, h_d)$. The set of Parikh vectors of paths of $\Gamma_{i,j}$ is denoted by $\Lambda(i, j)$. We consider the subset $\Lambda^{+h}(i, j)$ (resp. $\Lambda^{-h}(i, j)$) of Parikh vectors l in $\Lambda(i, j)$ such that $l \ge h$ (resp. $l \not\ge h$).

If there is no ambiguity, the following abbreviations are used: Λ_x, Λ_x^{+t}, Λ_x^{-t}; and similarly, Λ, Λ^{+h} and Λ^{-h}.

Proposition 7. *Let i and j be two vertices of Q. For all $x \in \Sigma$ such that $p_x \neq 0$, there exists an integer h_x such that $\Lambda_x^{+h_x}(i, j) = \{h_x + mp_x \mid m \ge 0\}$.*

Proof. Let $x \in \Sigma$. Let us consider the sequence $(l_1^x, \ldots, l_r^x)$, where l_i^x, $1 \le i \le r$, is the x-length of the simple cycle c_i, and the sub-sequence $(t_k^x)_{1\le k\le s}$ of its non-null elements.
Without loss of generality, we can assume that $t_1^x = \min(t_1^x, \ldots, t_s^x)$ and $t_s^x = \max(t_1^x, \ldots, t_s^x)$. Let us recall that $p_x = \gcd(t_1^x, t_2^x, \ldots, t_s^x)$.

Let us set $m_x = (\frac{t_1^x}{p_x} - 1)(\frac{t_s^x}{p_x} - 1)$. According to Corollary 3, for all integer $m \ge m_x$, mp_x is a linear combination with non-negative coefficients of the integers $l_1^x, \ldots, l_r^x$. Since $\mathcal{G}$ is strongly connected, according to Lemma 5 there exists a basic path $\gamma_0 \in \Gamma_{i,j}$ having a Parikh vector l_0, such that for any linear combination $S = \sum_{k=1}^{r} \alpha_k l_k$ of the Parikh vectors of the simple cycles (with non-negative coefficients) there exists a path of $\Gamma_{i,j}$ having a Parikh vector equal to $l_0 + S$.

Hence, for all $m \ge m_x$, there exists a path of $\Gamma_{i,j}$ having a x-length equal to $l_0^x + mp_x$. Setting $h_x = l_0^x + m_x p_x$, we obtain $\{h_x + mp_x \mid m \ge 0\} \subset \Lambda_x^{+h_x}$. Moreover, according to Lemma 6, two paths of $\Gamma_{i,j}$ have equal x-lengths modulo p_x. As a consequence, the set of x-lengths of paths of $\Gamma_{i,j}$ having a value greater than or equal to $h_x = l_0^x + m_x p_x$ is the set of integers $\{h_x + mp_x \mid m \ge 0\}$. $\square$

Lemma 8. *Let $\mathcal{G}$ be a d-ary strongly connected graph and $p = (p_1, \ldots, p_d)$ be its period. Let i be an arbitrarily chosen vertex of $\mathcal{G}$. For all $t = (t_1, \ldots, t_d) \in \{0, \ldots, p_1 - 1\} \times \cdots \times \{0, \ldots, p_d - 1\}$, let T_t be the subset of vertices j such that the Parikh vector of any path in $\Gamma_{i,j}$ is equal to t modulo p.*
Then the subsets T_t define a partition $\Pi(G)$ of the set Q of vertices of $\mathcal{G}$ such that for all $k \in \Sigma$, $T_{(t_1,\ldots,t_d)} \cdot k = T_{(t_1',\ldots,t_d')}$, with $t_k' = t_k + 1 \pmod{p_k}$ and, for all $m \neq k$, $t_m' = t_m$.

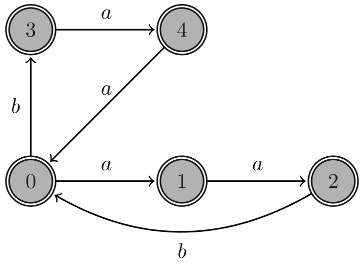

Fig. 3. The minimal automaton of L.

Table 1
The standard partition $\Pi_{\mathcal{G}}$.

(t_1, t_2)	$T_{(t_1,t_2)}$
(0, 0)	{0, 2, 3}
(1, 0)	{1, 4}

Proof. According to Lemma 6, the sets T_t, for $t = (0, \ldots, 0)$ to $t = (p_1 - 1, \ldots, p_d - 1)$, define a partition of Q such that $i \in T_{(0,\ldots,0)}$. For all $k \in \Sigma$, the set of k-successors of $T_{(t_1,\ldots,t_d)}$ is the set $T_{(t'_1,\ldots,t'_d)}$, with $t'_k = t_k + 1 \pmod{p_k}$ and, for all $m \neq k$, $t'_m = t_m$. □

Definition 9. (1) *The partition $\Pi(G)$ is the* standard partition *of $\mathcal{G}$.*
(2) *Two states j and j' that belong to a same class of $\Pi(G)$ are said to be* congruent.

Example 1. Let us consider the expression $E = (aab+baa)^*$ over the alphabet $\Sigma = \{a, b\}$. Let $\mathcal{A}$ be the minimal automaton of the language Pref$(L(E))$ (see Fig. 3).

The state diagram $\mathcal{G}$ of $\mathcal{A}$ has two simple cycles $c_1 = (0, 1, 2)$ and $c_2 = (0, 3, 4)$ such that $l_1 = l_2 = (2, 1)$. Therefore we have $p_a = 2$, $p_b = 1$ and $p = (2, 1)$. We consider the set $\Gamma_{(0,j)}$ of the paths that start from the initial state of $\mathcal{A}$. For $(t_1, t_2) \in \{(0, 0), (1, 0)\}$, $T_{(t_1,t_2)}$ is the subset of vertices j such that the Parikh vector of any path going from 0 to j is equal to (t_1, t_2) modulo p. The two sets $T_{(0,0)} = \{0, 2, 3\}$ and $T_{(1,0)} = \{1, 4\}$ define the standard partition $\Pi(G)$ (shown in Table 1) of the set Q of vertices of $\mathcal{G}$. We have: $T_{(0,0)} \cdot a = T_{(1,0)}$, $T_{(1,0)} \cdot a = T_{(0,0)}$ and $T_{(0,0)} \cdot a = T_{(0,0)} \cdot b = T_{(0,0)}$.

4. Geometry of strongly connected graphs

In this section, we focus on the set of successful paths of $\mathcal{A}$; we thus assume that the vertex i of $\mathcal{G}$ is the initial state s_0 of $\mathcal{A}$.

Definition 10. *Two distinct vertices j and j' are* compatible *($j \equiv j'$) if for any path $\gamma \in \Gamma_{i,j}$ and any path $\gamma' \in \Gamma_{i,j'}$, γ and γ' have a different Parikh vector. That is*

$$j \equiv j' \Leftrightarrow \Lambda(i, j) \cap \Lambda(i, j') = \emptyset.$$

Lemma 9. *If two vertices are not congruent, then they are compatible.*

Proof. As a consequence of Lemma 6, the Parikh vector l (resp. l') of any path $\gamma \in \Gamma_{i,j}$ (resp. $\gamma' \in \Gamma_{i,j'}$) is equal to some $t \in \mathbb{N}^d$ (resp. t') modulo p. It implies that, if $t \neq t'$ then $j \equiv j'$. □

Proposition 10. *Let L be a prefix closed regular language and $\mathcal{A}$ be its minimal automaton. We assume that the state diagram of $\mathcal{A}$ is a strongly connected graph. Then the language L is semi-geometrical if and only if the states of $\mathcal{A}$ are pairwise compatible.*

Proof. Let u and u' be two distinct words of L. We set $j = s_0 \cdot u$ and $j' = s_0 \cdot u'$. Since $\mathcal{A}$ is the deterministic minimal automaton of L, we have: $u^{-1}L = u'^{-1}L \Leftrightarrow \overrightarrow{L}_j^{\mathcal{A}} = \overrightarrow{L}_{j'}^{\mathcal{A}} \Leftrightarrow j = j'$. Hence the condition $c(u) = c(u') \Rightarrow u^{-1}L = u'^{-1}L$ of Definition 3 is equivalent to $c(u) = c(u') \Rightarrow j = j'$ (Condition 1). Let γ be a successful path in $\Gamma_{i,j}$ and γ' a successful path in $\Gamma_{i,j'}$. The condition $j \equiv j' \Leftrightarrow \Lambda(i, j) \cap \Lambda(i, j') = \emptyset$ of Definition 10 is equivalent to $l(\gamma) = l(\gamma') \Rightarrow j = j'$ (Condition 2). Let $w \in \Sigma^*$ be the label of a successful path γ of $\mathcal{A}$; then we have $l(\gamma) = c(w)$. By consequence, the Condition 1 and the Condition 2 are equivalent. □

Let us notice that state compatibility should only be checked on a deterministic automaton.

Definition 11. *Let x and y be two symbols in Σ such that $x < y$. Let j and j' be two distinct vertices of Q. The pair (j, j') is said to be a pair of (x, y)-neighbours if there exists a path $\gamma \in \Gamma_{i,j}$ and a path $\gamma' \in \Gamma_{i,j'}$ with respective Parikh vectors l and l', satisfying the following conditions:*
(1) $l^x = l'^x + 1$ *and* $l^y = l'^y - 1$,
(2) *For any symbol z different from x and different from y,* $l^z = l'^z$.

By definition, for a pair (j, j') of (x, y)-neighbours the condition $l + c(y) = l' + c(x)$ is satisfied.

Proposition 11. *Let L be a prefix closed regular language and $\mathcal{A}$ be its minimal automaton. We assume that the state diagram of $\mathcal{A}$ is a strongly connected graph. Then the language L is geometrical if and only if for every pair (j, j') of (x, y)-neighbours the condition $j \cdot y = j' \cdot x$ is satisfied.*

Proof. $\Rightarrow$ Let us show that if there exists a pair (j, j') of (x, y)-neighbours such that $j \cdot y \neq j' \cdot x$, then there exists a word $w \in \mathcal{L}(\mathcal{F}(L))$ that does not belong to L. Let u (resp. u') be the label of a successful path in $\Gamma_{i,j}$ (resp. $\Gamma_{i,j'}$). By definition u and u' are in L. Let us set $k = j \cdot y$ and $k' = j' \cdot x$. Note that at most one of the two states k and k' may be the sink state σ. By hypothesis, $k \neq k'$ and, since the DFA $\mathcal{A}$ is minimal, the right languages of k and k' are distinct sets. Without loss of generality let us asssume that the word w is in the right language of j and is not in the one of j'. Then $u'xw$ is not in L. On the opposite, since $c(uy) = c(u'x)$ and $uyw \in L$, it holds that $u'xw \in \mathcal{L}(\mathcal{F}(L))$.
$\Leftarrow$ Let us show that if L is not geometrical, then there exists a pair (j, j') of (x, y)-neighbours such that $j \cdot y \neq j' \cdot x$. By hypothesis, there exists a word $w \in \mathcal{L}(\mathcal{F}(L))$ that does not belong to L. Assuming that $L \neq \emptyset$, since $\varepsilon \in L$ we have $w \neq \varepsilon$. Then w can be decomposed as follows : $w = uyv$, with $y \in \Sigma$, $u \in L$ and $v \in \Sigma^*$, and u is the longest prefix of w that belongs to L. Since the word uy is in $\mathcal{L}(\mathcal{F}(L))$ there exists a word $u'x \in L$ such that $c(u'x) = c(uy)$. Let j (resp. j') be the tail of the successful path labelled by u (resp. u'). By construction, the pair (j, j') is a pair of (x, y)-neighbours such that $j \cdot y \neq j' \cdot x$. $\square$

In the case of a binary graph, with $\Sigma = \{a, b\}$, Proposition 11 becomes: a pair (j, j') is a *pair of neighbours* if there exists a path $\gamma \in \Gamma_{i,j}$ and a path $\gamma' \in \Gamma_{i,j'}$ with respective Parikh vectors l and l', such that $l^a = l'^a + 1$ and $l^b = l'^b - 1$. The language L is geometrical if and only if for every pair (j, j') of neighbours the condition $j \cdot b = j' \cdot a$ is satisfied.

4.1. Case of unary strongly connected graphs

Let L be a prefix closed unary regular language and $\mathcal{A}$ be its minimal automaton. Then, if L is an infinite language, we have $L = \Sigma^*$ and the state diagram of $\mathcal{A}$ reduces to a loop. Otherwise, $L = \{\varepsilon, a, aa, \ldots, a^t\}$ for some integer t and the state diagram is made of t components (each of them is reduced to one state). In both cases, the language L is semi-geometrical. Testing whether L is geometrical makes no sense since no vertex has a neighbour.

We now show that the geometry properties of a prefix closed unary regular language L can be characterized on any deterministic automaton $\mathcal{A}$ recognizing L. We assume that the state diagram $\mathcal{G}$ of $\mathcal{A}$ is a strongly connected unary graph.

The period of $\mathcal{C}$ is the integer $p = \gcd(l_1, \ldots, l_r)$. The set Λ (resp. Λ') is the set of lengths of paths in $\Gamma_{i,j}$ (resp. $\Gamma_{i,j'}$). The threshold h (resp. h') is an integer and Λ^{+h} (resp. $\Lambda'^{+h'}$) is the subset of Λ (resp. Λ') of the lengths that are greater than h (resp. h'). The path $\gamma_0 \in \Gamma_{i,j'}$ (resp. $\gamma'_0 \in \Gamma_{i,j'}$) is a basic path for the pair (i, j) (resp. (i, j')).

Lemma 12. *Let L be a prefix closed unary regular language and $\mathcal{A}$ be a deterministic automaton recognizing L. We assume that the state diagram of $\mathcal{A}$ is a strongly connected graph. Then the language L is semi-geometrical if and only if its standard partition is trivial.*

Proof. According to Lemma 9, we need only to check compatibility of congruent vertices. Let us show that any two congruent vertices j and j' are not compatible. According to Proposition 7, there exist two integers $h_0 = l_0 + m_0 p$ and $h'_0 = l'_0 + m_0 p$ such that $\Lambda^{+h_0} = \{h_0 + mp \mid m \geq 0\}$ and $\Lambda^{+h'_0} = \{h'_0 + mp \mid m \geq 0\}$. Since h_0 and h'_0 are equal modulo p, it holds that $\Lambda^{+h_0} \cap \Lambda^{+h'_0} \neq \emptyset$. Hence $\Lambda \cap \Lambda' \neq \emptyset$ and thus the two vertices j and j' are not compatible. $\square$

4.2. Case of d-ary strongly connected graphs

We now introduce two algorithms to check respectively the semi-geometricity and the geometricity of a d-ary prefix closed regular language for which the minimal automaton admits a strongly connected state diagram. These two algorithms are based on the resolution of systems of linear Diophantine equations. We first state a property of strongly connected graphs that is useful for proving the correctness of these algorithms.

Lemma 13. *Let us suppose that there exist a vector $m \in \mathbb{N}^d$ and a pair of paths (γ, γ') in $\Gamma_{i,j} \times \Gamma_{i,j'}$ such that $l(\gamma) = l(\gamma')$ and $l(\gamma) \not\geq m$. Then there exists a path μ in $\Gamma_{i,j}$ and a path μ' in $\Gamma_{i,j'}$ such that $l(\mu) = l(\mu')$ and $l(\mu) \geq m$.*

Proof. Since Σ is the alphabet of $\mathcal{A}$, for any x in Σ and $\mathcal{A}$ is strongly connected, there exists at least an edge of $\mathcal{G}$ that is labelled by x. It implies that there exists a (non necessarily simple) cycle C going through i and j such that, for any x in Σ, at least one edge is labelled by x. Hence the Parikh vector λ of C is such that, for all x in Σ, $\lambda^x \geq 1$. Let us consider the path μ in $\Gamma_{i,j}$ made of the path γ followed by k turns through the cycle C and the path μ' in $\Gamma_{i,j'}$ made of k turns through the cycle C followed by the path γ'. It is obvious that the two paths μ and μ' have the same Parikh vector and that moreover this length can be made greater than m by choosing a large enough k. As a consequence, if there exists a pair of paths having a same Parikh vector not greater than m then there exists a pair of paths having a same Parikh vector greater than m. $\square$

4.2.1. Semi-geometricity test

Proposition 14. *Let L be a d-ary prefix closed regular language and $\mathcal{A}$ be its minimal automaton; the state diagram of $\mathcal{A}$ is assumed to be a strongly connected graph and the vertex i is assumed to be the initial state of $\mathcal{A}$. Let (j, j') be a pair of distinct vertices, γ_0 (resp. γ'_0) be a basic path from i to j (resp. j') and $(l_0^1, \ldots, l_0^d)$ (resp. $({l'}_0^1, \ldots, {l'}_0^d)$) be the Parikh vector of γ_0 (resp. γ'_0).*

With the pair (j, j') is associated the following system $\mathcal{S}(j, j')$ of linear Diophantine equations (with $\alpha_1, \ldots, \alpha_r, \alpha'_1, \ldots, \alpha'_r$ as unknowns):

$$(l_0^1, \ldots, l_0^d) + \sum_{1 \le k \le r} \alpha_k (l_k^1, \ldots, l_k^d) = ({l'}_0^1, \ldots, {l'}_0^d) + \sum_{1 \le k \le r} \alpha'_k (l_k^1, \ldots, l_k^d).$$

Then the language L is semi-geometrical if and only if, for any pair (j, j') of congruent vertices, the set of solutions of the system $\mathcal{S}(j, j')$ is empty.

Proof. According to Lemma 9, we only need to check compatibility for pairs (j, j') of congruent states. Let us set $l_0 = (l_0^1, \ldots, l_0^d)$, $l'_0 = ({l'}_0^1, \ldots, {l'}_0^d)$ and $m = \max(l_0, l'_0)$. There exists a solution $(\alpha_1, \ldots, \alpha_r, \alpha'_1, \ldots, \alpha'_r)$ for the system $\mathcal{S}(j, j')$ if and only if there exist a path γ in $\Gamma_{i,j}$ and a path γ' in $\Gamma_{i,j'}$ such that the Parikh vectors $l = l_0 + \sum_{1 \le k \le r} \alpha_k l_k$ of γ and $l' = l'_0 + \sum_{1 \le k \le r} \alpha'_k l_k$ of γ' are equal. By construction we have $l \ge m$ and thus the following condition holds:

$$\Delta = \emptyset \Leftrightarrow \Lambda^{+m}(i, j) \cap \Lambda^{+m}(i, j') = \emptyset.$$

According to Lemma 13, we have:

$$\Lambda^{+m}(i, j) \cap \Lambda^{+m}(i, j') = \emptyset \Leftrightarrow \Lambda^{-m}(i, j) \cap \Lambda^{-m}(i, j') = \emptyset.$$

Finally the following condition is satisfied:

$$\Delta = \emptyset \Leftrightarrow \Lambda(i, j) \cap \Lambda(i, j') = \emptyset$$

and according to Proposition 10 the language L is semi-geometrical if and only if, for any pair (j, j') of congruent vertices, the set of solutions of the system $\mathcal{S}(j, j')$ is empty. □

The complexity of the algorithm depends on the number of pairs of congruent vertices. According to Lemma 8 the standard partition contains $p_1 \cdots p_d$ classes, and thus in the case where $p_1 = \ldots = p_d = 1$ there is a unique class containing n elements. As a consequence, in the worst case, there are $O(n^2)$ systems to be solved.

4.2.2. Geometricity test

We denote by $\mathbf{u}_i$ the element of $\mathbb{N}^d$ whose components are all null except the i-th one, which equals 1.

Proposition 15. *Let L be a d-ary prefix closed regular language and $\mathcal{A}$ be its minimal automaton. The state diagram of $\mathcal{A}$ is assumed to be a strongly connected graph and the vertex i is assumed to be the initial state of $\mathcal{A}$. Let (j, j') be a pair of distinct vertices, γ_0 (resp. γ'_0) be a basic path from i to j (resp. j') and $(l_0^1, \ldots, l_0^d)$ (resp. $({l'}_0^1, \ldots, {l'}_0^d)$) be the Parikh vector of γ_0 (resp. γ'_0). With the tuple $(j, j', x, y) \in Q \times Q \times \Sigma \times \Sigma$, such that $x < y$, is associated the following system $\mathcal{S}'(j, j', x, y)$ of linear Diophantine equations (with $\alpha_1, \ldots, \alpha_r, \alpha'_1, \ldots, \alpha'_r$ as unknowns):*

$$(l_0^1, \ldots, l_0^d) + \sum_{1 \le k \le r} \alpha_k (l_k^1, \ldots, l_k^d) + \mathbf{u}_y = ({l'}_0^1, \ldots, {l'}_0^d) + \sum_{1 \le k \le r} \alpha'_k (l_k^1, \ldots, l_k^d) + \mathbf{u}_x.$$

Then the language L is geometrical if and only if, for all (j, j', x, y) with $x < y$ such that the system $\mathcal{S}'(j, j', x, y)$ admits a solution, the condition $j \cdot y = j' \cdot x$ is satisfied.

Proof. According to Proposition 11, the language L is geometrical if and only if for every pair (j, j') of (x, y)-neighbours the condition $j \cdot y = j' \cdot x$ is satisfied. Let us set $m = \max(l_0, l'_0)$. Given a tuple (j, j', x, y) with $j \neq j'$ and $x < y$, there exists a solution $(\alpha_1, \ldots, \alpha_r, \alpha'_1, \ldots, \alpha'_r)$ for the system $\mathcal{S}(j, j', x, y)$ if and only if there exist a path γ in $\Gamma_{i,j}$ and a path γ' in $\Gamma_{i,j'}$ with respective Parikh vector $l = l_0 + \sum_{1 \le k \le r} \alpha_k l_k$ and $l' = l'_0 + \sum_{1 \le k \le r} \alpha'_k l_k$ such that $l + \mathbf{u}_y = l' + \mathbf{u}_x$, that is such that (j, j') is a pair of (x, y)-neighbours. As a consequence, the system $\mathcal{S}(j, j', x, y)$ allows us to detect any pair (j, j') of (x, y)-neighbours such that the condition $l \ge l_0 \wedge l' \ge l'_0$ is satisfied.

Let us suppose now that there exists a path γ in $\Gamma_{i,j}$ and a path γ' in $\Gamma_{i,j'}$ such that $l + \mathbf{u}_y = l' + \mathbf{u}_x$, with $l \ngeq l_0$ or $l' \ngeq l'_0$. According to Lemma 13 there exist a path μ in $\Gamma_{i,j}$ and a path μ' in $\Gamma_{i,j'}$ with Parikh vectors $l(\mu)$ and $l(\mu')$ such that $l(\mu) + \mathbf{u}_y = l(\mu') + \mathbf{u}_x$, with $l(\mu) \ge l_0$ and $l(\mu') \ge l'_0$. Moreover, the condition $j \cdot y = j' \cdot x$ is not changed when going from paths γ and γ' to paths μ and μ'. As a consequence, if the condition $j \cdot y = j' \cdot x$ is satisfied for every pair (j, j') of (x, y)-neighbours corresponding to a solution of the system $\mathcal{S}(j, j', x, y)$, then it is satisfied for every pair (j, j') of (x, y)-neighbours. □

Table 2
The basic paths $\gamma_0(0, j)$.

j	1	2	3	4
$\gamma_0(0, j)$	(0, 1)	(0, 1, 2)	(0, 3)	(0, 3, 4)
The label of $\gamma_0(0, j)$	a	aa	b	ba
$l(\gamma_0(0, j))$	(1, 0)	(2, 0)	(0, 1)	(1, 1)

There are at most $O(d^2n^2)$ systems to be solved. In the case of a binary graph, Proposition 15 is simplifed as follows.

Corollary 16. *Let $\Sigma = \{a, b\}$ (with $a < b$) and $(j, j') \in Q \times Q$. Let us consider the following system $\mathcal{S}'(j, j')$ of linear Diophantine equations:*

$$(l_0^a, l_0^b) + \sum_{1 \le k \le r} \alpha_k (l_k^a, l_k^b) = (l'^a_0, l'^b_0) + \sum_{1 \le k \le r} \alpha'_k (l_k^a, l_k^b) + (1, -1).$$

Then the language L is geometrical if and only if for all (j, j'), the condition $j \cdot b = j' \cdot a$ is satisfied whenever the set of solutions of the system $\mathcal{S}'(j, j')$ is nonempty.

Example 2. Let us consider the expression $E = (aab + baa)^*$ defined in the Example 1. Let us check whether the language $L = \mathrm{Pref}(L(E))$ is semi-geometrical or whether it is geometrical. Let $\mathcal{A}$ be the minimal automaton of L (see Fig. 3).

Let us recall that there are two simple cycles $c_1 = (0, 1, 2)$ and $c_2 = (0, 3, 4)$ such that $l_1 = l_2 = (2, 1)$ and that the standard partition $\Pi_{\mathcal{G}}$ is shown in Table 1. For $1 \le j \le 4$, Table 2 shows the the basic path $\gamma_0(0, j)$ that is considered, as well as its label and its Parikh vector $l(\gamma_0(0, j))$.

Let us show that the language L is semi-geometrical. According to Table 1, there are two pairs of congruent states: (2, 3) and (1, 4). Since the two simple cycles c_1 and c_2 have the same Parikh vector, we can make use of a unique variable α_1 instead of two variables α_1 and α_2. The system $\mathcal{S}_{(2,3)}$: $(2, 0) + \alpha_1(2, 1) = (0, 1) + \alpha'_1(2, 1)$ has no solution. Therefore the states 2 and 3 are compatible. Similarly the system $\mathcal{S}_{(1,4)}$: $(1, 0) + \alpha_1(2, 1) = (0, 1) + \alpha'_1(2, 1)$ has no solution. Therefore the states 1 and 4 are compatible. By consequence, the language L is semi-geometrical.

The system $\mathcal{S}'_{(1,3)}$: $(1, 0) + \alpha_1(2, 1) = (0, 1) + \alpha'_1(2, 1) + (1, -1)$ has a nonempty set of solutions. Moreover $1 \cdot b = \sigma \ne 4 = 3 \cdot a$. By applying Corollary 16, one immediately gets that the language L is not geometrical.

5. Experimental study

An application has been developed [3] in order to check the new method. Let L be the language to be tested and $\mathcal{A}$ be the minimal automaton of L. Actually, the data is a regular expression denoting L that is converted into a finite automaton; this automaton is then determinized and minimized. Let $\mathcal{G}$ be the state diagram of $\mathcal{A}$. The graph $\mathcal{G}$ is first partitioned into strongly connected components. The method being used thus far is limited to a graph reduced to a unique strongly connected component. Let us assume that $\mathcal{G}$ is reduced to such a graph. The list of the simple cycles of $\mathcal{G}$ is computed, as well as its standard partition. Then, if we consider for example the test of semi-geometricity, for all the pairs (j, j') of confluent states, the system $\mathcal{S}(j, j')$ is constructed and given as a data to the algorithms of the Polylib library [25]. As soon as the solver finds a solution for one of these systems, the language L is declared to be not semi-geometrical. A precise analysis of the performance of each step of this method is in progress. Yet, it appears that the resolution of Diophantine equation systems is a greedy step.

6. Conclusion

We have introduced a new method for testing whether a regular language is semi-geometrical or not and whether it is geometrical or not, in the case where the minimal automaton of L admits a strongly connected diagram. We intend to deepen the theoretical relation that exists with the method described in [1] which also handles arbitrarily large alphabets, as well as with the geometrical solution obtained for the 2-dimensional case [4]. For both tests, we have to consider a set of systems of linear Diophantine equations; actually we need not know the solutions of these systems, only the existence of a solution. Recent studies address the problem of proving that there is no solution [8,17] and the LinBox library [13] is expected to provide a response to this question. We intend to use this library in our future experiments.

References

[1] B. Blanpain, J.-M. Champarnaud, J.Ph. Dubernard, Geometrical languages, in: C. Martin-Vide (Ed.), International Conference on Language Theory and Automata, LATA'07, in: GRLMC Universitat Rovira I Virgili, vol. 35/07, 2007, pp. 127–138.
[2] A. Brauer, On a problem of partitions, Amer. J. Math 64 (1942) 299–312.
[3] F. Carpentier, Systèmes d'équations diophantiennes et test de géométricité sur un langage rationnel. Rapport de master, Université de Rouen, France, 2008.
[4] J.-M. Champarnaud, J.Ph. Dubernard, H. Jeanne, Geometricity of binary regular languages, in: A.-H. Dediu, H. Fernau, C. Martín-Vide (Eds.), International Conference on Language and Automata Theory and Applications, LATA 2010, in: Lecture Notes in Computer Science, vol. 6031, 2010, pp. 178–189.

[5] J.-M. Champarnaud, J.Ph. Dubernard, H. Jeanne, Regular geometrical languages and tiling the plane, in: M. Domaratzki, K. Salomaa (Eds.), CIAA, in: Lecture Notes in Computer Science, vol. 6482, Springer, 2010, pp. 69–78.
[6] J.-M. Champarnaud, G. Hansel, Puissances des matrices booléennes. Unsubmitted manuscript, Université de Rouen, France, 2005.
[7] M. Chrobak, Finite automata and unary languages, Theor. Comput. Sci. 47 (3) (1986) 149–158.
[8] F. d'Alessandro, B. Intrigila, S. Varricchio, On some counting problems for semi-linear sets. CoRR, abs/0907.3005, 2009.
[9] S. Eilenberg, Automata, Languages and Machines, vol. B., Academic Press, New York, 1976.
[10] J. Geldenhuys, B. van der Merwe, L. van Zijl, Reducing nondeterministic finite automata with SAT solvers, in: A. Yli-Jyrä, A. Kornai, J. Sakarovitch, B.W. Watson (Eds.), FSMNLP, in: Lecture Notes in Computer Science, vol. 6062, Springer, 2009, pp. 81–92.
[11] D. Geniet, G. Largeteau, WCET free time analysis of hard real-time systems on multiprocessors: a regular language-based model, Theor. Comput. Sci. 388 (1–3) (2007) 26–52.
[12] S. W. Golomb, Polyominoes: Puzzles, Patterns, Problems, and Packings, Princeton Academic Press, 1996.
[13] LinBox Group. Linbox project : exact computational linear algebra. http://www.linalg.org, 2002.
[14] J. C. Holladay, R. S. Varga, On powers of non negative matrices, Proc. Amer. Math. Soc. 9 (4) (1958) 631–634.
[15] S. Kleene, Representation of events in nerve nets and finite automata, Automata Studies, Ann. Math. Studies 34 (1956) 3–41.
[16] L. Mordell, Diophantine Equations, Academic Press, London, 1969.
[17] T. Mulders, A. Storjohann, Certified dense linear system solving, J. Symb. Comput. 37 (4) (2004) 485–510.
[18] J. Myhill, Finite automata and the representation of events. WADD, TR-57-624:112–137, 1957.
[19] A. Nerode, Linear automata transformation. In Proceedings of AMS 9, 1958, pp. 541–544.
[20] T. Paranthoën, Génération aléatoire et structure des automates à états finis. Ph.D., Université de Rouen, France, 2004.
[21] R. Parikh, On context-free languages, J. ACM 13 (4) (1966) 570–581.
[22] R. W. Robinson, et al., Counting strongly connected finite automata, in: Y. Alavi (Ed.), Graph Theory with Applications to Algorithms and Computer Science, Wiley, New York, 1985, pp. 671–685.
[23] A. Schrijver, Theory of Linear and Integer Programming, John Wiley and Sons, New York, 1986.
[24] R. E. Tarjan, Enumeration of the elementary circuits of a directed graph, SIAM J. Comput. 2 (1973) 211–216.
[25] The Polylib Team. Polylib User's Manual. http://www.irisa.fr/polylib/doc/, IRISA, France, 2002.